PATENT APPLICATION COVER SHEET
Attorney Docket No. 3408.69858

*I hereby certify that this paper is being deposited with the United States Postal Service as EXPRESS MAIL in an envelope addressed to: MS Patent Application, Commissioner for Patents, Alexandria, VA 22313-1450, on this date.*

*2/27/04* *Date* *Express Mail No. EV032736882US*

# WRITE PROCESSING METHOD FOR STREAM TYPE COMMANDS AND MEDIUM STORAGE APPARATUS

## INVENTORS:

Hiroaki MURAI
Yoshinori INOUE
Satoru FUKASE
Takehiro OGAWA

GREER, BURNS & CRAIN, LTD.
300 South Wacker Drive
Suite 2500
Chicago, Illinois 60606
Telephone: 312.360.0080
Facsimile: 312.360.9315
CUSTOMER NO. 24978

# WRITE PROCESSING METHOD FOR STREAM TYPE COMMANDS AND MEDIUM STORAGE APPARATUS

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application is based upon and claims the benefit of priority from the prior Japanese Patent Application No. 2003-131859, filed on May 9, 2003, the entire contents of which are incorporated herein by reference.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates to a write processing method for stream type commands and a medium storage apparatus for writing data on the medium in accordance with stream type commands, and more particularly to a write processing method for stream type commands and a medium storage apparatus for writing data continuously within a specified time.

### 2. Description of the Related Art

Because of the recent advancements in data processing technology, the processing of moving pictures, music and voice is becoming possible. Because of this, data storage apparatus has begun to be used to store such data. For this application, stream type commands, such as Read Stream/Write Stream, are provided.

A stream type command is primarily used for AV (Audio Visual). Therefore priority is assigned to performance (speed) rather than to the original reliability of the data storage apparatus (e.g. magnetic disk apparatus).

In a conventional data storage apparatus, if a write error is generated when data is written to the medium, normally a retry is performed and a read/write is attempted until the threshold is reached. In the case of a stream type command, where the requesting unit continuously transmits data, however, if an error occurs, processing including a retry must be completed within a specified time. Therefore conventionally, if a write error occurs, the command is terminated as an error.

However, there are some cases where such an error termination is not preferable as the application of data storage apparatus expands. For example, in the case of real-time picture recording and sound recording, a re-transmission cannot be expected even if the command is terminated as an error.

Also in some cases, sectors, where data cannot be correctly written to the medium, may be existed, and such sectors cannot be used even if they are read.

## SUMMARY OF THE INVENTION

With the foregoing in view, it is an object of the present invention to provide a write processing method for

stream type commands and a medium storage apparatus for writing correct data without terminating the command as an error, and with maintaining the execution time, even if a write error occurs during a data write by the stream type commands.

It is another object of the present invention to provide a write processing method for stream type commands and a medium storage apparatus for reading data correctly when data is read, without terminating as an error, and with maintaining the execution time, even if a write error occurs during a data write by the stream type commands.

To achieve these objects, a write processing for stream type commands of the present invention for writing write data to a storage medium by a head according to the stream type commands includes steps of: writing received write data of the stream type commands to a buffer; writing the write data of the buffer sequentially to the storage medium by the head while confirming whether the writing succeeded; skipping sectors where the writing did not succeed when the writing did not succeed in the write step, and protecting the write data in the sectors where the writing did not succeed on the buffer; and writing the write data of the protected sector to the storage medium by the head after the stream type commands are executed.

The medium storage apparatus of the present invention for writing data to a storage medium by a head includes: a buffer for storing write data received along with stream type

commands; a controller for sequentially writing the write data of the buffer to the storage medium by the head; and a processing unit for confirming whether the writing succeeded, and skipping sectors where the writing did not succeed and protecting the write data of the sector where the writing did not succeed on the buffer when the writing did not succeed. And the processing unit writes the write data of the protected sectors to the storage medium by the head after executing the stream type commands.

According to the present invention, with the write stream type commands, writing to error sectors is skipped so that processing, which normally ends if an error occurs, is continued, and skipped sectors are written after command execution, so the error is corrected such that the medium is correctly read in the next and later read processing, while maintaining the write time specified by the command.

It is preferable that the present invention further includes a step of restarting the write processing of the head after stopping the write processing to the storage medium when the writing did not succeed. By this, the error status of the head can be recovered and writing can be continued.

In the present invention, it is preferable that the protect step includes a step of protecting the write data of the sector where the writing did not succeed and the sector for which writing was skipped until restart. By this, even if writing is stopped and then the writing is restarted, the

write data for which writing was skipped can be protected.

In the present invention, it is preferable that the step of writing the data of the protected sector includes a step of enabling a predetermined number of times of retries when the writing of the write data did not succeed. By this, writing becomes more certain since the retry is enabled.

In the present invention, it is preferable that the protect step further includes a step of storing the sector where the writing did not succeed, and the address of the buffer of the write data of the sector in a protect table. Because of recording to a protect table, the sector for which writing was skipped can be recognized later, even when the write processing is continued.

Also in the present invention, it is preferable that the step of writing the write data of the protected sector includes a step of writing the write data of the buffer in reference to the protect table. By this, the sector for which writing was skipped can be written easily.

Also in the present invention, it is preferable that the protect step includes a step of calculating a shortest start sector in terms of the time up to the restart after the write processing is stopped, and a step of protecting the write data of the sector where the writing did not succeed and the sector for which writing was skipped up to the restart obtained from the calculation result. By this, the sector for which writing was skipped can be easily detected.

Also in the present invention, it is preferable that the

storage medium is a rotating disk medium. By this, the present invention can be applied to a large capacity medium.

Also in the present invention, it is preferable that the protect step includes a step of recognizing that the command is a stream type command and executing the protection when the writing did not succeed. By this, the write processing for a stream type command can be executed using a normal write processing routine.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram depicting the medium storage apparatus according to an embodiment of the present invention;

Fig. 2 is a diagram depicting a task file in Fig. 1;

Fig. 3 is a diagram depicting a feature register in Fig. 2;

Fig. 4 shows the table of the memory in Fig. 1;

Fig. 5 shows a detailed configuration of the table in Fig. 4;

Fig. 6 is a diagram depicting the data buffer in Fig. 1;

Fig. 7 is a flow chart depicting write processing according to an embodiment of the present invention (No. 1);

Fig. 8 is a flow chart depicting write processing according to an embodiment of the present invention (No. 2);

Fig. 9 is a flow chart depicting write processing according to an embodiment of the present invention (No. 3);

Fig. 10 is a diagram depicting the operation of processing in Fig. 7 to Fig. 9;

Fig. 11 is a diagram depicting an example of the stream type write commands in the processing in Fig. 7 to Fig. 9;

Fig. 12 shows the command table of the stream type commands example in Fig. 11; and

Fig. 13 shows a protect table of the stream type write commands example in Fig. 11.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the present invention will now be described in the sequence of medium storage apparatus, data write processing of stream type commands, and other embodiments.

[Medium storage apparatus]

Fig. 1 is a block diagram depicting the medium storage apparatus according to an embodiment of the present invention, Fig. 2 is a diagram depicting a task file in Fig. 1, Fig. 3 is a diagram depicting a feature register in Fig. 2, Fig. 4 shows the table of the memory in Fig. 1, Fig. 5 shows the configuration of the table in Fig. 4, and Fig. 6 is a diagram depicting the data buffer in Fig. 1.

Fig. 1 shows a magnetic disk apparatus (HDD) for reading/writing data from/to the magnetic disk as a medium storage apparatus. The magnetic disk apparatus 10 is connected with a host, such as a personal computer, via an

ATA (AT Attachment) standard interface cable 9.

The magnetic disk apparatus 10 is comprised of a magnetic disk 19, a spindle motor 20 for rotating the magnetic disk 19, a magnetic head 25 for reading/writing data from/to the magnetic disk 19, and an actuator (VCM) for moving the magnetic head 25 in the radius direction (track crossing direction) of the magnetic disk 19.

The control section is comprised of an HDC (Hard Disk Controller) 26, data buffer 14, MPU 11, memory 13, head IC 18, spindle motor driver 21, VCM driver 23, position detection section 24, and a bus 17 which connects all these components.

The HDC 26 is further comprised of an ATA interface control circuit 12 which has a task file 12A for setting a task from the host, data buffer control circuit 15 for controlling the data buffer 14, and a format control circuit 16 for controlling the format of the recording data.

The head IC 18 supplies recording current to the magnetic head 25 according to the recording data when writing, or amplifies the read signal from the magnetic head 25 and outputs the read data (including servo information) when reading. The position detection section 24 detects the position of the magnetic head 25 using the servo information from the head IC 18.

The spindle driver 21 drives the rotation of the spindle motor 20. The VCM driver 23 drives the VCM 22 for moving the magnetic head 25. The MPU (Micro-Processor) 11 performs position control of the magnetic head 25, read/write control,

and retry control. And the memory 13 stores tables and data required for processing of the MPU 11.

Fig. 2 is a diagram depicting the ATA standard task file (register) 12A in Fig. 1, where the command string (task) from the host is set. In the command register 1F7, commands such as read and write commands are set. The start address is set in the apparatus/head register 1F6, and in the cylinder high, low and sector number registers 1F5 - 1F3. The number of sectors for read/write is set in the sector count register 1F2.

In the feature register 1F1, the mode of read or write is specified. As Fig. 3 shows, bit 7 of the 8-bit feature register 1F1 is the URGENT (URG) bit, and bit 6 is the WRITE CONTINUE (WC) bit. The URG bit specifies urgent processing, and this specifies, for example, to inhibit retry and to execute write operation in the case of a write command. The WC bit specifies to continue writing and specifies a write mode where only several times of retry is enabled.

In the present embodiment, a stream type command is specified by specifying the URG bit or the WC bit.

Fig. 4 shows a table which is set for the memory 13 in Fig. 1, Fig. 5 shows a configuration of the table in Fig. 4, and Fig. 6 is a diagram depicting the data buffer 14 in Fig. 1. As Fig. 4 shows, the command table 13A and the protect table 13B are created in the memory 13.

As Fig. 6 shows, in the data buffer 14, write data from the host or read data to the host are stored at each buffer

address position in LBA units.

As Fig. 5 shows, the command table 13A stores the command received from the host, the start LBA (Logical Block Address), the sector count and the later mentioned protect bit. The protect table 13B stores the command number to be protected, the start LBA of the area to be protected in the data buffer 14 in Fig. 6, the sector count number to be protected, the address of the data buffer 14 and the protect bit.

[Data write processing of stream type commands]

Now data write processing when a stream type command is received in the above mentioned configuration will be described using the processing flow charts in Fig. 7 to Fig. 9.

(S10) The MPU 11 judges whether a command from the host via HDC 26 is not received for a predetermined time. When no command is received from the host for a predetermined time, processing advances to step S32 in Fig. 9.

(S12) The MPU 11 judges whether the HDC 26 received a command from the host. If the command is not received, processing returns to step S10.

(S14) When the MPU 11 receives a command via the HDC 26, the MPU 11 analyzes the command code and sends an instruction corresponding to the command to the HDC 26. In other words, when the MPU 11 judges that this command is not a Write Stream PIO (Port Input/Output) command by the command code analysis, the MPU 11 sends normal processing to the HDC 26.

In the normal processing, importance is placed on the reliability of the data, so a predetermined number of times of retries is performed for the read/write error regardless the time, so that reliability of the data is guaranteed.

(S16) When the MPU 11 judges that the command is a Write Stream PIO command, the MPU 11 updates the command table 13A on the memory 13 according to the command which is set in the task file 12A of the HDC 26 (see Fig. 2). The write data which is sent from the host is stored to the data buffer 14 (see Fig. 6) via the HDC 26 (ATA interface control circuit 12, buffer control circuit 15).

(S18) Then the MPU 11 instructs the operation of the Write Stream PIO command to the HDC 26. By this, the buffer control circuit 15 of the HDC 26 drives the magnetic head 25 via the format circuit 16 and the head IC 18 by the received write data on the data buffer 14, and writes the received write data on the buffer 14 to the magnetic disk 19.

(S20) The MPU 11 detects the decoded position from the position detection section 24, drives the VCM 22 via the VCM driver 23, and controls the position of the magnetic head 25 to the specified track. The MPU 11 also judges whether the magnetic head has deviated from the track center in a predetermined range (off track error) using the decoded position.

(S22) When the MPU 11 judges that a write error, such as an off track error, occurred, the MPU 11 stops the medium write processing of the HDC 26 at that point (at the point

when the write error occurred). Since the command at this time is with an URG bit (Urgent bit) in the feature register 1F1 (see Fig. 3), the MPU 11 does not attempt a retry, and calculates the shortest start LBA in terms of time up to restart. In other words, even if the write operation is stopped due to a write error, the magnetic disk 19 is rotating, so the predetermined number of sectors (LBA) cannot be written until restart (return of the magnetic head 25 to the track center by VCM 22 and confirmation thereof).

(S24) The MPU 11 updates the protect table 13B (see Fig. 4) on the memory 13. In other words, the MPU 11 adds to the protect table 13B the start LBA which could not be written, the sector count number which could not be written, calculated from the shortest start LBA mentioned above, the buffer addresses thereof in the buffer 14 and the protect bit (protective flag). In other words, data on the erred sectors and the sectors for which writing was skipped up to restart are protected.

(S26) If MPU 11 calculated that restart from LBA 11 is possible, as described later in Fig. 10, the MPU 11 starts medium write processing to the HDC 26 again from the LBA 11, without performing any processing for LAB 5 - 10. And processing returns to step S18. In Fig. 10, a write error is detected at LBA 4, so LBAs which could not be written or have not yet been written are LBA 4 - LBA 10.

(S28) If a write error is not detected in step S20, the MPU 11 judges whether a command end notice was received from

the HDC 26. If there is no command end notice, processing returns to step S18.

(S30) When a command end notice is received from the HDC 26, the MPU 11 updates the command table 13A (see Fig. 5) on the memory 13. In other words, the protect bit of the command table 13A is changed from "1" to "0". And processing returns to step S10 in Fig. 7. After this, the MPU 11 does not use buffer area of the buffer addresses 0x0600-0x13FF, even if a new command is issued from the host, because the buffer addresses 0x0600 - 0x13FF (see Fig. 6) on the buffer 14 which stores data are protected by the protect table 13B (see Fig. 5).

(S32) On the other hand, it is judged in step S10 whether the host did not issue a command for a while after this. For example, it is judged whether a command was not issued for one minute. If a command was not issued for a predetermined time, the MPU 11 instructs the medium write processing again to the HDC 26 based on the protect table 13B (see Fig. 5) on the memory 13. Since this is not a command issued by the host, the HDC 26 performs a full retry, attempting to complete the medium writing as much as possible. In other words, when the MPU 11 judges that a write error, such as an off track error, occurred, the MPU 11 stops medium write processing to the HDC 26 at that point (at the point when the write error occurred), and attempts retry.

(S34) Then the MPU 11 judges whether a retry out occurred (error continues after a predetermined number of

retries).

(S36) If a retry out did not occur, that is if the medium writing completed, the MPU 11 receives a write completion from the HDC 26. When the write completion is received, the MPU 11 ends the medium write processing via the HDC, and updates the protect table 13B in Fig. 5. In other words, the MPU 11 clears the executed protect table 13B, and returns to the command reception wait status, which is a normal sequence in step S10.

(S38) If medium writing failed in step S34 even after a full retry, the MPU 11 gives up writing the LBA and moves to writing the next LBA. For example, if a full retry medium writing to the LBA 4 failed, the MPU 11 gives up writing the LBA 4 and moves to a full retry medium writing for the next LBA 5. If it is judged that the writing has not been completed in step S36, the MPU 11 moves to writing the next LBA. And processing returns to step S32.

In other words, in the flow in Fig. 9, the steps from S32 to S38 are repeated for the number of the sector count written in the protect table 13B. When all these steps are completed, the protect table 13B in Fig. 5 is updated, and the protection of the buffer data 14 is cleared. When the next and later commands are received, the MPU 11 performs operation for using the buffer addresses 0x0600 - 0x13FF (see Fig. 6).

This will be described in detail using a specific example with reference to Fig. 10 to Fig. 13. It is assumed

that the commands (No. 0, No. 1 . . . ) are issued from the host in the command sequence shown in Fig. 11. In this case, the MPU 11 updates the command table 13A on the memory 13 (see Fig. 12).

The write data sent from the host is stored to the buffer 14 via the HDC 26, as shown in Fig. 6. The MPU 11 judges this as a Write Stream PIO, and instructs this information to the HDC 26, and the HDC 26 writes the received write data on the buffer 14 to the medium 19.

The commands No. 0 and No. 1 in Fig. 11 are written to the medium without problem, and the normal end status is returned to the host. If a write error (e.g. off track error) occurs to the No. 2 command thereafter, the HDC 26 stops the medium write processing at that point (at the point when the write error occurred). The command this time is an URG bit (Urgent bit) in the feature register, so a retry is not attempted and the MPU 11 calculates the shortest start LBA in terms of time up to restart.

The MPU 11 updates the protect table 13B on the memory 13, as shown in Fig. 13. In other words, as the protect table 13B at the bottom of Fig. 13 shows, the start LBA which could not be written, the sector count, the buffer address and the protect bit are added.

As Fig. 10 shows, if the write error is detected at LBA 4 and the MPU 11 calculated that restart is possible from LBA 11, then nothing is executed (writing is skipped) for LBA 5 to LBA 10, and the medium writing is started from the LBA 11

via the HDC. In this case, LBAs which could not actually be written or have not yet been written are LBA 4 to LBA 10 in Fig. 10.

When the command ends, the MPU 11 updates the command table 13A on the memory 13, and changes the protect bit from "1" to "0" (see Fig. 12). Then the buffer area on which the protect bit is changed are protected by the protect table 13B (see Fig. 13) so that the buffer addresses 0x0600 to 0x13FF on the buffer 14 (Fig. 6), which stores the data, are not used even if a new command is issued by the host.

If the host does not issue a command for a while after this, for example does not issue a command for one minute, the MPU 11 performs medium write processing again for the buffer data in Fig. 6, which has been protected up to this point, based on the protect table 13B (see Fig. 13) on the memory 13. In this case, a command was not issued by the host, so a fully retry is attempted, so as to complete the medium writing as much as possible.

If the medium writing succeeded at this point, the MPU 11 ends the medium write processing via the HDC, updates the protect table 13B in Fig. 13, and returns to command reception wait status, which is a normal sequence.

If medium write still fails after the full retry, the MPU 11 gives up writing the LBA and moves to writing the next LBA. For example, if the full retry medium writing to LBA 4 failed, the MPU 11 gives up writing LBA 4, and moves to a full retry medium writing for the next LBA 5.

This step is repeated for the number of sector counts written in the protect table 13B (see Fig. 13). When all these steps are completed, the MPU 11 updates the protect table 13B in Fig. 13, and clears the protection of the buffer data (Fig. 6). When the protect table 13B becomes this status, the MPU 11 performs operation using the buffer addresses 0x0600 – 0x13FF (Fig. 6) when the next and later commands are received.

In this way, according to the present invention, processing which ends normally if an error occurs can be continued in the Write Stream related commands, and the error can be corrected such that the medium is correctly read in the next or later read processing.

Also when no command is issued from the host for a predetermined time, the sectors for which writing was skipped are written, so writing can be executed without using the host. Also the protect table is created, and sectors for which writing was skipped are stored and are protected on the buffer, so the loss of data from the buffer is prevented, and write processing can be executed for certain after a command ends.

In rewriting after a command ends, a full retry is possible, so the writing of sectors for which writing was skipped becomes more certain.

[Other embodiments]

In the above embodiments, the medium was described using a magnetic disk, but the present invention can be applied to

other storage media, such as optical disks and magneto-optical disks. The write error was described as an off track error, but the present invention can be applied to other write errors. And the interface is not limited to an ATA, but may be other interfaces.

Moreover, as described in the example of the Urgent bit, the above embodiments can be achieved likewise also in the WRITE CONTINUE bit which affords for several retries. Furthermore, a protect area may be provided separately in the buffer to copy the protected data.

The present invention was described using embodiments, but the present invention can be modified in various forms within the scope of the essential character of the present invention, which shall not be excluded from the scope of the present invention.

In this way, according to the present invention, for the Write Stream commands, processing which ends normally if an error occurs is continued with skipping the writing of the erred sectors, and the data of the erred sectors is protected and the sectors, for which writing was skipped, are written after command execution. So the error can be corrected such that the medium is correctly read in the next and later read processing while maintaining the write time specified by the command.